PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-116261
(43)Date of publication of application : 06.05.1998

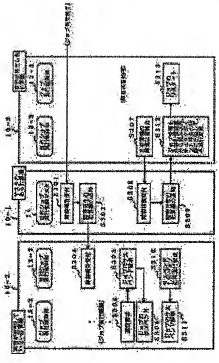
(51)Int.Cl. G06F 15/16
G06F 9/46
G06F 11/14

(21)Application number : 08-270874
(22)Date of filing : 14.10.1996

(71)Applicant : HITACHI LTD
(72)Inventor : AOYANAGI YUKIKO
SAITO YOSHIMICHI

(54) CHECK POINT RESTARTING METHOD FOR PARALLEL COMPUTER SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To shorten the execution time of all parallel jobs after a restart when a fault has occurred at an arbitrary computer which executes the parallel jobs.
SOLUTION: A parallel job execution control master 11 once receiving a fault report on a computer 10-3 informs a computer 10-2, where no fault has occurred of the fault. The computer 10-2 continuous to execute the job as it is and interrupts the job when a request to communicate with the faulty computer 10-3 is made. Then when the computer 10-3 recovers from the fault, the parallel job execution control master 11 informs the computer 10-2 of that. In response, the computer 10-2 restarts the interrupted job.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報(A)　(11)特許出願公開番号

特開平10－116261

(43)公開日　平成10年(1998)５月６日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　15/16	４７０	Ｇ０６Ｆ　15/16	４７０Ｒ
9/46	３６０	9/46	３６０Ｚ
11/14	３１０	11/14	３１０Ｂ

審査請求　未請求　請求項の数３　ＯＬ　（全９頁）

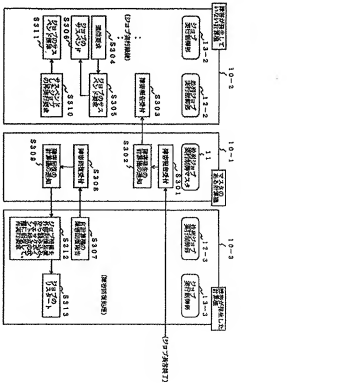
(21)出願番号　特願平8－270874

(22)出願日　平成８年(1996)10月14日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　青柳　由紀子
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(72)発明者　斉藤　喜道
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(74)代理人　弁理士　鈴木　誠

(54)【発明の名称】　並列計算機システムのチェックポイントリスタート方法

(57)【要約】
【課題】　並列ジョブを実行する任意の計算機で障害が発生した時に、リスタート後の並列ジョブ全体の実行時間を短縮する。
【解決手段】　並列ジョブ実行制御マスタ１１は、計算機１０－３の障害報告を受けると、それを障害の発生していない計算機１０－２に通知する。計算機１０－２は、ジョブの実行をそのまま継続し、障害の発生した計算機１０－３に対して通信要求が発生すると、当該ジョブの実行を中断する。その後、計算機１０－３の障害が回復すると、並列ジョブ実行制御マスタ１１は、それを計算機１０－２に通知する。これを受けて、計算機１０－２は、中断していた当該ジョブの実行を再開する。



1

【特許請求の範囲】
【請求項１】　複数の計算機がネットワークで結合され、ジョブを複数の計算機に振り分けて実行する並列計算機システムにおいて、各々の計算機が所定の時点毎にジョブの実行情報を当該計算機の具備する外部記憶装置に格納し、障害が発生した時に、前記格納した情報を元にジョブの状態を障害発生前の時点に回復してジョブの実行を再開するチェックポイントリスタート方法であって、
任意の計算機で障害が発生した時に、障害の発生していない計算機は、ジョブを引き続いて実行して、前記障害の発生した計算機に対して通信要求や当該計算機資源に対するアクセス要求（以下、通信要求で総称する）が生じた時点でジョブの実行をサスペンド状態にし、
前記障害の発生した計算機が障害を回復すると、前記障害の発生していない計算機は、前記サスペンド状態を解除してジョブの実行を再開する、ことを特徴とする並列計算機システムのチェックポイントリスタート方法。
【請求項２】　複数の計算機がネットワークで結合され、ジョブを複数の計算機に振り分けて実行する並列計算機システムにおいて、各々の計算機が所定の時点毎にジョブの実行情報を当該計算機の具備する外部記憶装置に格納し、障害が発生した時に、前記格納した情報を元にジョブの状態を障害発生前の時点に回復してジョブの実行を再開するチェックポイントリスタート方法であって、
任意の計算機で障害が発生した時に、障害の発生していない計算機は、ショブを引き続いて実行して、前記障害の発生した計算機に対して通信要求が生じた時点で異常終了とし、
前記障害の発生した計算機が障害を回復すると、前記障害の発生していない計算機は、前記異常終了させたジョブに対応する実行情報を外部記憶装置から読み出し、ジョブの状態を異常終了前の時点に回復してジョブの実行を再開する、ことを特徴とする並列計算機システムのチェックポイントリスタート方法。
【請求項３】　請求項１もしくは２記載の並列計算機システムのチェックポイントリスタート方法において、障害の発生した計算機が障害を回復し、ジョブの実行を再開して、障害の発生していない計算機に通信要求を出すと、当該障害の発生していない計算機が、サスペンド状態の解除もしくは異常終了の回復処理を行うことを特徴とする並列計算機システムのチェックポイントリスタート方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は並列計算機システムにおけるジョブの実行制御方法に係り、特に障害発生時の並列ジョブのチェックポイントリスタート方法に関する。

2

【０００２】
【従来の技術】一般に計算機システムでは、処理の過程で障害が発生し、続いて処理ができなくなった場合に備えて、所定の時点、時点でジョブの実行情報を外部記憶装置に格納し、障害が発生した場合、該格納した情報を元に、ジョブの状態を障害発生前の当該格納した時点に回復してジョブの実行を再開する機能を備えている。このジョブの実行情報を外部記憶装置に格納する時点をチェックポイントと称し、障害が発生した場合に、その直前のチェックポイントで格納した情報を元に、当該チェックポイントからジョブの実行を再開することをチェックポイントリスタートと称する。
【０００３】従来、複数の計算機がネットワークで結合され、並列ジョブを各計算機に振り分けて実行する並列計算機システムにおけるチェックポイントリスタートでは、並列ジョブが実行する任意の計算機で障害が発生すると、そのジョブに関わる他の計算機上の並列ジョブも異常終了し、障害が発生した計算機で障害を取り除いて運用が再開すると、各々の計算機は、当該計算機の具備する外部記憶装置に格納した当該ジョブに対応する実行情報を元に、格納時の実行状態に回復して並列ジョブの実行を再開していた。
【０００４】
【発明が解決しようとする課題】従来技術では、並列ジョブ実行中の任意の計算機で障害が発生すると、障害の起きていない他の計算機上の並列ジョブも直ちに異常終了して処理を中断するので、障害が発生してから処理を再開するまでの間、障害の起きていない計算機に無駄な空き時間が生じる問題があった。また、障害のない計算機上の並列ジョブの処理に時間がかかるような場合、障害が回復してから処理を再開すると、回復後の並列ジョブ全体の処理実行時間が長くなる問題があった。さらに、アベンドしたジョブをリスタートする場合、外部記憶装置から該当するジョブの実行情報を取得して回復するので、その分、処理時間が長くかかる問題があった。
【０００５】本発明の主たる目的は、並列計算機システムにおける並列ジョブのチェックポイントリスタートにおいて、障害の起きていない計算機の無駄な空き時間をなくし、障害回復後の並列ジョブ全体の実行時間の短縮を図ることにある。
【０００６】
【課題を解決するための手段】請求項１の発明は、並列ジョブを実行する任意の計算機で障害が発生した時に、障害の発生していない計算機は、ジョブを引き続いて実行し、前記障害の発生した計算機と通信要求や資源アクセス要求（以下、通信要求で総称する）が生じた時点でジョブの実行をサスペンド状態（中断）にし、障害の発生した計算機が障害を回復したなら、前記障害の発生していない計算機はサスペンド状態を解除して、当該ジョブの実行を再開するようにしたことである。

【０００７】これにより、ある計算機で障害が発生した時に、障害の発生していない計算機上のジョブを出来る限り停止させずに処理を継続させることができ、障害回復後の並列ジョブ全体の実行時間を短縮することができる。特に、障害のない計算機上の並列ジョブの処理に時間がかかる場合に有効である。さらに、該請求項１の発明では、障害が回復した時、障害のない計算機では、ジョブのサスペンド状態を解除して当該ジョブの処理を再開するだけでよく、外部記憶装置からジョブ実行情報を取得する必要がなく、その分の処理時間も短縮できる。

【０００８】請求項２の発明は、並列ジョブを実行する任意の計算機で障害が発生した時に、障害の発生していない計算機は、ジョブを引き続いて実行して、前記障害の発生した計算機に対して通信要求が生じた時点で異常終了とし、障害の発生した計算機が障害を回復したなら、前記障害の発生していない計算機は、異常終了させたジョブに対応する実行情報を外部記憶装置から読み出し、ジョブの状態を異常終了前の時点（チェックポイント）に回復してジョブの実行を再開するようのしたことである。

【０００９】これにより、請求項１の発明と同様に、ある計算機で障害が発生した時に、障害の発生していない計算機上のジョブを出来る限り停止させずに継続させることができ、障害の回復後の並列ジョブ全体の実行時間を短縮することができる。さらに、該請求項２の発明では、障害の発生していない計算機は、障害の発生した計算機に対して通信要求が生じた時点で異常終了とすることで、障害の回復が長びくような場合に、該障害の発生していない計算機を一旦、電源オフとし、障害回復の報告を受けて再立上げする（この時、異常終了直前のチェックポイントから処理が再開する）ことが可能になり、障害の発生していない計算機の無駄な稼動状態を回避できる。また、この間、障害の発生していない計算機の保守・診断も可能になる。

【００１０】請求項３の発明は、請求項１の発明の障害の発生していない計算機のサスペンド状態の解除や請求項２の発明の障害の発生していない計算機の異常終了の回復処理を、障害の発生した計算機が障害を回復し、並列ジョブの実行を再開して、障害の発生していない計算機に通信要求を出したことを契機に行うとしたことである。これにより、障害の発生していない計算機では、障害の発生した計算機から通信要求を受け取るまで、他のジョブの処理に専念することが可能になる。

【００１１】

【発明の実施の形態】以下、本発明の一実施例について図面により詳細に説明する。図１は本発明を適用した並列計算機システムの概略構成を示すブロック図である。図において、複数台の計算機１０がネットワーク１００にそれぞれ接続され、並列計算機システムを構成している。図１では、計算機１０は４台だけを示しているが、計算機１０の台数は任意である。各計算機１０は、図示しないＣＰＵやメモリなどから構成され、それぞれ外部記憶装置２０を具備している。なお、ＣＰＵは、各計算機１０に一つあるいはそれ以上あってもよい。計算機監視装置３０は各計算機１０の稼働状態を監視するものである。図１では、該計算機監視装置３０は、各計算機１０と同様にネットワーク１００に接続されるとしたが、それぞれ個別の信号線で各計算機１０と接続してもよい。

【００１２】初めに、図２により、このような並列計算機システムにおける並列ジョブの一般的な実行制御について説明する。なお、図２では、１台の計算機に対応する外部記憶装置しか示していないが、図１に示したように、各計算機毎に外部記憶装置が存在することは云うまでもない。

【００１３】並列計算機システムを構成する各計算機１０には、図２に示すように、並列ジョブ実行制御部１２とジョブ実行制御部１３がある。ジョブ実行制御部１３は、当該計算機上で動作するジョブのスケジューリングやスワッピング、割込み処理等といったジョブの実行制御を行なう。並列ジョブ実行制御部１２は、当該計算機上で動作する並列ジョブの管理を行い、ジョブ実行制御部１３の下で実行するように制御する。また、並列ジョブ実行制御部１２は、他の計算機上で動作する並列ジョブとの通信や、同期等の制御手段を提供する。並列ジョブ実行制御マスタ１１は、各計算機上で動作する並列ジョブ実行制御部１２を一括して管理し、並列ジョブの起動や、並列ジョブの分割、並列ジョブの振り分けを行う。該並列ジョブ実行制御マスタ１１は、並列計算機システムを構成する複数の計算機１０上の何れかで動作する。図２では、１つの計算機１０で並列ジョブ実行制御部に替えて、専用に並列ジョブ実行制御マスタ１１が動作している場合を示している。他には、ひとつの計算機１０上に並列ジョブ実行制御部１２と並列ジョブ実行制御マスタ１１があってもよい。

【００１４】並列計算機システムを構成する各計算機１０のジョブ実行制御部１３は、並列ジョブの実行の途中で、チェックポイント毎に、当該計算機のレジスタ値やスタック情報、データ情報などといったジョブのリスタートに必要な情報（ジョブ実行情報２１）を外部記憶装置２０に格納する。チェックポイントの契機は、並列ジョブ実行制御部１２により与えられても、ジョブ自身がジョブ中に設定したコードにより与えてもよい。ジョブをリスタートする場合、ジョブ実行制御部１３は、当該計算機の外部記憶装置２０に格納されている当該ジョブに対応するジョブ実行情報２１を読み出して、その情報からジョブを当該チェックポイントの時点の状態に回復し、処理を再開する。

【００１５】計算機監視装置３０は、各計算機１０の稼動状態を監視し、いずれかの計算機で障害が発生する

と、当該装置３０より並列ジョブ実行制御マスタ１１に通知する。また、逆に並列ジョブ実行制御マスタ１１が計算機監視装置３０に、所定時間隔で各計算機１０の稼動状態を問い合わせてもよい。並列ジョブ実行制御マスタ１１は、計算機監視装置３０によより障害の発生した計算機を知ると、当該並列ジョブ実行制御マスタ１１が管理する各計算機１０の並列ジョブ実行制御部１２に障害の発生した計算機を通知する。本発明は、この後のジョブ実行制御にかかわる。

【００１６】次に、図３乃至図５により、本発明の並列ジョブチェックポイントリスタート制御の二、三の実施例について説明する。なお、以下の説明では、並列ジョブ実行制御マスタのある計算機を１０－１、障害の発生していない計算機を１０－２、障害の発生した計算機を１０－３とする。同様に、障害の発生していない計算機１０－２の並列ジョブ実列制御部を１２－２、ジョブ実行制御部を１３－２とし、また、障害の発生した計算機１０－３の並列ジョブ実行制御部を１２－３、ジョブ実行制御部を１３－３とする。

【００１７】図３は、本発明の並列ジョブにおけるチェックポイントリスタートの処理の第１の実施例を示すフローチャートで、請求項１の発明に対応するものである。並列ジョブを実行中にある計算機１０－３で障害が発生し、運用が不可能になると、計算機１０－１の並列ジョブ実行制御マスタ１１は、計算機監視装置３０により該障害の発生した計算機１０－３を知る（ステップＳ３０１）。並列ジョブ実行制御マスタ１１は、該障害が発生して運用が不可能になった計算機１０－３を、該並列ジョブ実行制御マスタ１１が管理する各並列ジョブ実行制御部１２に通知する（ステップＳ３０２）。障害の発生していない計算機１０－２の並列ジョブ実行制御部１２－２は、該並列ジョブ実行制御マスタ１１からの障害の報告を受け取る（ステップＳ３０３）。この間、障害の発生した計算機１０－３では、障害の回復を試みている。一方、障害の発生していない計算機１０－２のジョブ実行制御部１３－２は、ジョブの処理を継続する。そして、当該ジョブが障害の発生した計算機１０－３と通信を行なうために、ジョブ実行制御部１３－２が並列ジョブ実行制御部１２－２に通信要求を出すと（ステップＳ３０４）、並列ジョブ実行制御部１２－２は、ステップＳ３０３で通信対象である計算機１０－３の障害報告を受け付けているので、通信要求のあったジョブのサスペンド要求をジョブ実行制御部１３－２に依頼する（ステップＳ３０５）。これを受けて、ジョブ実行制御部１３－２は当該ジョブをサスペンド状態（一時停止状態）にする（ステップＳ３０６）。

【００１８】その後、障害の発生した計算機１０－３で障害が取り除かれると、並列ジョブ実行制御部１２－３は障害の回復を計算機１０－１の並列ジョブ実行制御マスタ１１に報告する（ステップＳ３０７）。並列ジョブ実行制御マスタ１１は、計算機１０－３の障害回復報告を受け取ると（ステップＳ３０８）、該並列ジョブ実行制御マスタ１１が管理する各並列ジョブ実行制御部１２に障害回復を通知する（ステップＳ３０９）。なお、ステップ３０８では、並列ジョブ実行制御マスタ１１が計算機監視装置３０により計算機１０－３の障害回復を知ってもよい。障害の発生していない計算機１０－２の並列ジョブ実行制御部１２－２は、並列ジョブ実行制御マスタ１１から計算機１０－３の障害回復の通知を受け取ると、サスペンドした並列ジョブを再開するように、ジョブ実行制御部１３－２に依頼する（ステップＳ３１０）。これにより、ジョブ実行制御部１３－２は、当該ジョブのサスペンドを解除し、該ジョブの実行を再開する（ステップＳ３１１）。一方、障害が回復した計算機１０－３の並列ジョブ実行制御部１２－３では、ステップ３０９で並列ジョブ実行制御マスタ１１から、障害回復報告を受け取ると、障害発生直前の当該ジョブで実行していた並列ジョブに対応するジョブ実行情報を外部記憶装置２０から読み込み、当該ジョブ実行情報を格納したチェックポイント時点からジョブをリスタートするようにジョブ実行制御部１３－３に依頼する（ステップＳ３１２）。これにより、ジョブ実行制御部１３－３はジョブのリスタートを行なう（ステップＳ３１３）。

【００１９】図４は、本発明の並列ジョブにおけるチェックポイントリスタートの処理の第２の実施例を示すフローチャートで、請求項２の発明に対応するものである。障害の発生していない計算機１０－２の並列ジョブ実行制御部１２－２が、並列ジョブ実行制御マスタ１１から計算機１０－３の障害発生の報告を受け取り、ジョブ実行制御部１３－２がジョブの処理を継続するところまでは、図３の第１の実施例と同様である。該障害の発生していない計算機１０－２のジョブ実行制御部１３－２上で実行する並列ジョブが、障害の発生した計算機１０－３と通信を行なうために通信要求を出した時（ステップＳ４０１）、当該計算機上の並列ジョブ実行制御部１２－２は、当該ジョブの異常終了の要求をジョブ実行制御部１２－２に出す（ステップＳ４０２）。これを受けて、ジョブ実行制御部１３－２は当該ジョブを異常終了させる（ステップＳ４０３）。その後、障害の発生した計算機１０－３の障害が取り除かれて、並列ジョブ実行制御部１２－３から障害回復が報告され（ステップＳ４０４）、並列ジョブ実行制御マスタ１１で受け付けられると（ステップＳ４０５）、並列ジョブ実行制御マスタ１１は、当該並列ジョブ実行制御マスタ１１が管理する各並列ジョブ実行制御部１２に障害回復を通知する（ステップＳ４０６）。この通知を受けて、障害の発生していない計算機１０－２の並列ジョブ実行制御部１２－２および障害の発生した計算機１０－３の並列ジョブ実行制御部１２－３は、各々、当該ジョブで実行していた並列ジョブに対応するジョブ実行情報を自外部記憶装置２

０から読み込み、当該情報を格納したチェックポイント時点からジョブをリスタートするようにジョブ実行制御部１３－２、１３－３に依頼する（ステップＳ４０７）。これを受けて、ジョブ実行制御部１３－２、１３－３は、各々ジョブのリスタートを行う（ステップＳ４０８）。これにより、障害の発生していない計算機１０－２の並列ジョブは、ステップＳ４０３で異常終了した直前のチェックポイントから処理を再開し、障害の発生した計算機１０－３の並列ジョブは、障害発生の直前のチェックポイントから処理を再開する。

【００２０】図５は、本発明の並列ジョブにおけるチェックポイントリスタートの処理の第３の実施例を示すフローチャートである。これは、並列ジョブを実行中にある計算機１０－３で障害が発生し、障害の発生していない計算機１０－２上で実行している並列ジョブが、障害の発生している計算機１０－３と通信を行なうために通信要求を出し、当該障害の発生していない計算機１０－２がサスペンド状態或は異常終了した後の、ジョブ再開処理の他の実施例を示したフローで、請求項３の発明に対応するものである。

【００２１】障害の発生した計算機１０－３で障害が取り除かれ、計算機１０－１の並列ジョブ実行制御マスタ１１に障害回復の報告が通知されると（ステップＳ５０１、Ｓ５０２）、並列ジョブ実行制御マスタ１１は、障害があった計算機１０－３上で動作する並列実行制御部１２－３に、障害回復を通知する（ステップＳ５０３）。なお、並列ジョブ実行制御マスタ１１は、該ステップＳ５０３で、当該並列実行制御マスタ１１が管理する各並列ジョブ実行制御部１２に障害回復の通知をし、該障害回復の通知を受け取った並列ジョブ実行制御部１２は、自計算機の障害回復かどうか判断して、自計算機でない場合は障害回復通知を無視するようにしてもよい。障害の発生した計算機１０－３の並列ジョブ実行制御部１２－３は、障害回復を通知されると、当該計算機１０－３で実行していた並列ジョブに対応するジョブ実行情報を自外部記憶装置２０から読み込み、当該情報を格納したチェックポイント時点からジョブをリスタートするようにジョブ実行制御部１３－３に依頼する（ステップＳ５０４）。これを受けてジョブ実行制御部１３－３はジョブのリスタートを行う（ステップＳ５０５）。

【００２２】リスタート後、当該ジョブが他の計算機１０－２上の並列ジョブと通信を行なうために、ジョブ実行制御部１３－３が並列ジョブ実行制御部１２－３に通信要求を出すと（ステップＳ５０６）、これを受けた並列ジョブ実行制御部１２－３は、並列実行制御マスタ１１に並列ジョブの通信要求を出す（ステップＳ５０７）。これを受けて、並列ジョブ実行制御マスタ１１は、通信対象となる計算機１０－２上の並列ジョブ実行制御部１２－２にジョブの再実行を要求する（ステップＳ５０８）。これにより、並列ジョブ実行制御部１２－２はジョブ実行制御部１３－２にジョブの再実行を指示し（ステップＳ５０９）、ジョブの実行を再開する（ステップＳ５１０）。この場合、ジョブ実行制御部１３－２は、並列ジョブがサスペンドして中断中の場合は、図３に示したように、サスペンド状態を解除して当該ジョブの実行を再実行する。また、当該ジョブが異常終了して停止している場合は、図４に示したように、当該ジョブに対応するジョブ実行情報を自外部記憶装置２０から読み込み、当該情報を格納したチェックポイント時点からジョブをリスタートする。

【００２３】以上、本発明の並列ジョブチェックポイントリスタート処理の二、三の実施例について説明したが、これらの実施例は障害の種類や度合等で使い分けてもよい。例えば、障害が軽微で比較的短時間に回復する場合には図３に示す第１の実施例を適用し、致命的な障害で、回復に長時間かかる場合には図４に示す第２の実施例を適用すればよい。この場合、計算機監視装置３０が、障害発生した計算機とともに、その障害の種類や度合等を並列ジョブ実行制御マスタに通知し、これを並列ジョブ実行制御マスタが自分の管理する各並列ジョブ実行制御部に連絡し、当該並列ジョブ実行制御部がいずれのケースを選択するか判断すればよい。

【００２４】なお、本発明では、並列ジョブを実行する任意の計算機で障害が発生した場合、障害の発生していない計算機では、ジョブの実行をそのまま継続し、該障害の発生した計算機に対して通信要求が生じた時点で、当該ジョブをサスペンドあるいは異常終了とするため、当該ジョブの実行が再開されると、障害の回復した計算機に対して、あらためて通信要求を出すことになる。一方、障害の回復した計算機では、障害発生前のチェックポイントからジョブの実行が再開されるため、障害の発生していない計算機より処理が遅れ、通信要求に対して正しい応答を返せない場合がある。このような場合には、障害の発生していない計算機は、正しい応答が返るまで通信要求を繰り返すようにすればよい。これにより、障害の発生していない計算機では、後続の処理が待たされることとなるが、このようなケース（正しい応答を返せないケース）は頻繁にある訳ではなく、ほとんど支障はない。

【００２５】

【発明の効果】以上説明したように、本発明によれば、並列計算機システムにおける並列ジョブのチェックポイントリスタート処理において、障害の発生した計算機の障害を取り除いている間に、障害の発生していない計算機のジョブを、障害の発生した計算機と通信や資源のアクセスを行なうまで継続して処理することで、障害回復後の並列ジョブ全体の実行時間を短縮することができる。さらに、請求項１の発明では、障害が回復した時、障害の発生していない計算機は、ジョブのサスペンド状態を解除して当該ジョブの処理を再開するだけでよく、

外部記憶装置から当該ジョブの実行情報を取得する必要がなく、その分の処理時も短縮できる。また、請求項２の発明では、障害の発生していない計算機は、障害の発生した計算機に対して通信要求等が生じた時点で異常終了とすることで、障害の回復が長びくような場合には、一旦、電源オフとして、障害回復の報告をまって再立上げすることが可能になり、無駄な稼動状態を回避でき、また、この間、当該計算機の保守・診断も可能になる。また、請求項３の発明では、障害の発生していない計算機のジョブの再開を、障害の発生した計算機が障害を回復して、並列ジョブの実行を再開し、当該障害の発生していない計算機に通信要求を出したことを契機とすることで、その間、障害の発生していない計算機は他のジョブの処理に専念することができ、計算機のさらなる有効利用が可能になる。

【図面の簡単な説明】

【図１】本発明の一実施例を示すシステム構成のブロック図である。

＊【図２】並列計算機システムにおける並列ジョブの実行制御を説明する図である。

【図３】本発明の並列ジョブにおけるチェックポイントリスタート処理の第１の実施例を示すフローチャートである。

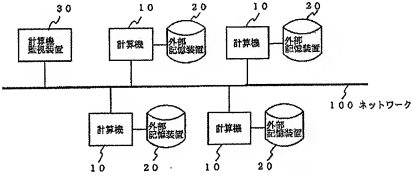
【図４】本発明の並列ジョブにおけるチェックポイントリスタート処理の第２の実施例を示すフローチャートである。

【図５】本発明の並列ジョブにおけるチェックポイントリスタート処理の第３の実施例を示すフローチャートである。

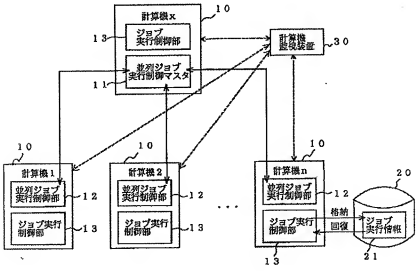
【符号の説明】

１０　計算機
１１　並列ジョブ実行制御マスタ
１２　並列ジョブ実行制御部
１３　ジョブ実行制御部
２０　外部記憶装置
３０　計算機監視装置

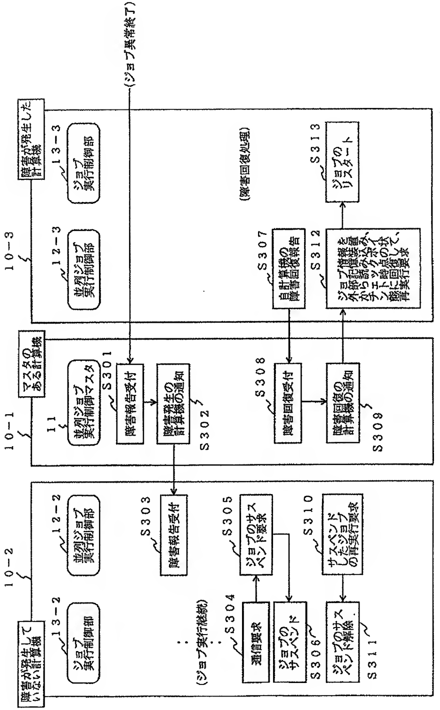
【図１】



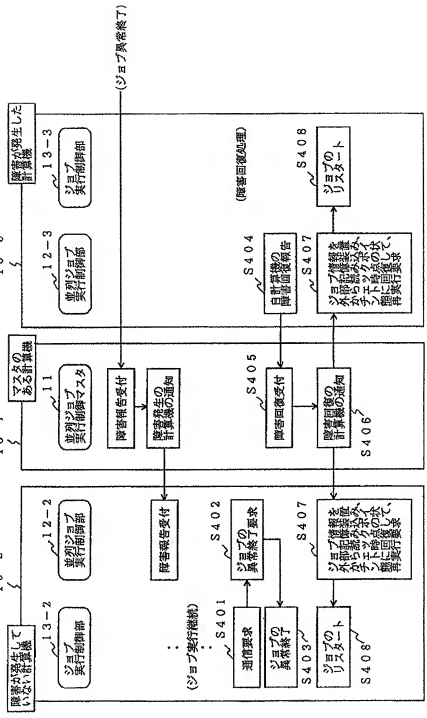
【図２】



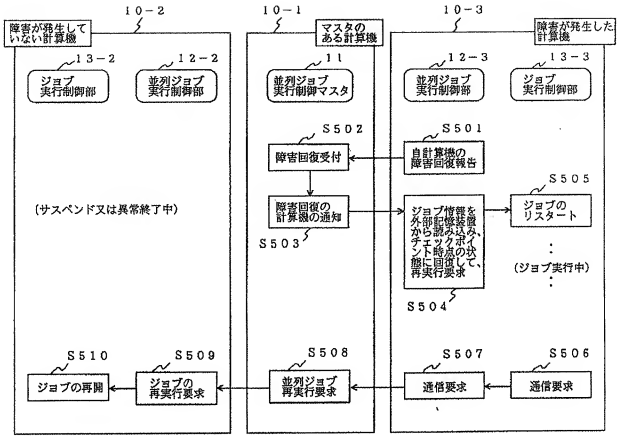
【図3】


障害が発生した計算機
10－3
12－3
並列ジョブ実行制御部
13－3
ジョブ実行制御部
(ジョブ異常終了)
(障害回復処理)
S307
自計算機の障害回復報告
S312
ジョブ情報を外部記憶装置から読み込み、チェックポイント時点の状態に回復して、再実行要求
S313
ジョブのリスタート
マスタのある計算機
10－1
11
並列ジョブ実行制御マスタ
S301
障害報告受付
S302
障害発生の計算機の通知
S308
障害回復受付
S309
障害回復の計算機の通知
障害が発生していない計算機
10－2
12－2
並列ジョブ実行制御部
13－2
ジョブ実行制御部
S303
障害報告受付
(ジョブ実行継続)
S304
通信要求
S305
ジョブのサスペンド要求
S306
ジョブのサスペンド
S310
サスペンドしたジョブの再実行要求
S311
ジョブのサスペンド解除

【図4】


障害が発生していない計算機
10－2
13－2
ジョブ実行制御部
12－2
並列ジョブ実行制御部
障害が発生した計算機
10－3
13－3
ジョブ実行制御部
12－3
並列ジョブ実行制御部
マスタのある計算機
10－1
11
並列ジョブ実行制御マスタ
(ジョブ異常終了)
障害報告受付
障害報告の計算機への通知
(ジョブ実行抑制)
S401
通信要求
S402
ジョブの異常終了要求
S403
ジョブ異常終了
S404
自計算機の障害報告
S405
障害回復受付
S406
障害回復の計算機への通知
S407
ジョブ情報を最新のものに更新し、ジョブ再実行要求
S408
ジョブのリスタート
(障害回復処理)

【図5】


10－2
障害が発生していない計算機
13－2
ジョブ実行制御部
12－2
並列ジョブ実行制御部
（サスペンド又は異常終了中）
S510
ジョブの再開
S509
ジョブの再実行要求
10－1
マスタのある計算機
11
並列ジョブ実行制御マスタ
S502
障害回復受付
障害回復の計算機の通知
S503
S508
並列ジョブ再実行要求
10－3
障害が発生した計算機
12－3
並列ジョブ実行制御部
13－3
ジョブ実行制御部
S501
自計算機の障害回復報告
ジョブ情報を磁気装置等の外部記憶装置から読み込み、チェックポイント時点の状態に回復して、再実行要求
S504
S505
ジョブのリスタート
（ジョブ実行中）
S507
通信要求
S506
通信要求